

MULTIMEDIA



UNIVERSITY

STUDENT ID NO

--	--	--	--	--	--	--	--	--	--

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 1, 2022/2023 (TERM ID 2210)

ECE3136 – JAVA TECHNOLOGY

(All Sections / Groups)

13 FEBRUARY 2023
2:30 P.M – 4:30 P.M.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This examination paper consists of SIX (6) pages including cover with THREE (3) questions only.
2. Answer ALL questions. All questions carry 25 marks and the distribution of the marks for each question is given.
3. This examination is OPEN-BOOK. Allowed material: Textbook and/or bound notes only.
4. Please write all your answers in the Answer Booklet provided.

Question 1

The UML Class diagram is a graphical notation used to construct and visualize object-oriented program. Based on the UML class diagram in *Figure Q1.1*, convert it into Java code. [Note: No driver program is required]

Write the **Printable** interface.

[1 marks]

Write the **Document** abstract class

[7 marks]

Write the **TextFile** class which inherits the **Document** class and implements the **Printable** interface

[8 marks]

Write the **SpreadsheetFile** class which inherits the **Document** class and implements the **Printable** interface

[9 marks]

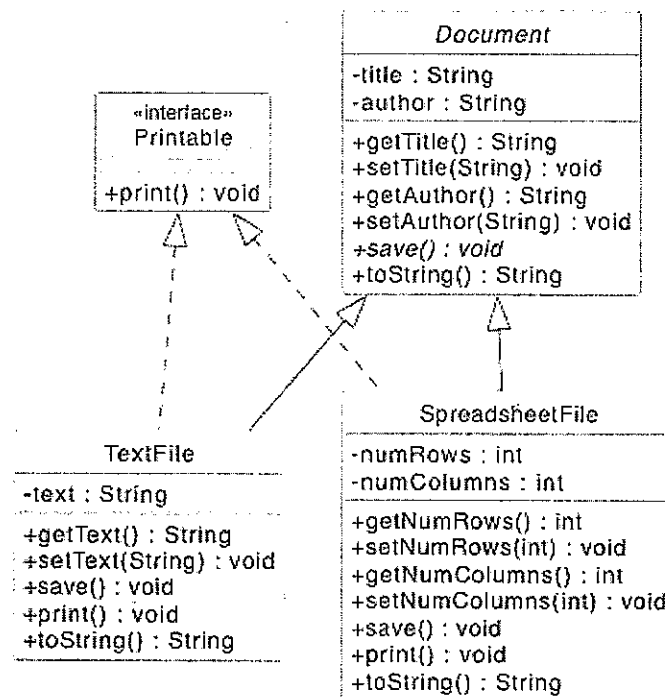


Figure Q1.1

Continued...

Question 2

Write a program with a graphical user interface (GUI) for ferry vehicle load checking.

Write the abstract class, **Vehicle**. Include the attributes **numberPlate** and **weight**, where **numberPlate** is to store the vehicle number plate, and **weight** is to store the weight of the vehicle. Provide a parameterized constructor which calls the mutator methods. Besides that, also provide all the necessary accessor and mutator methods, and the **toString** method. The **toString** method should return the vehicle number plate and the weight. Also, provide an abstract method **type** which returns a String.

[4 marks]

Write a class, **Car**, which inherits from **Vehicle**. Provide a constructor, and all the necessary accessor and mutator methods. Override the superclass abstract method **type** to return the type of the vehicle, which is "Car" in this case. Also, override the superclass **toString** method by appending the vehicle type right before the superclass **toString** return value.

[4 marks]

Write another class, **Truck**, which inherits from **Vehicle**. Provide a constructor, and all the necessary accessor and mutator methods. Override the superclass abstract method **type** to return the type of the vehicle, which is "Truck" in this case. Also, override the superclass **toString** method by appending the vehicle type right before the superclass **toString** return value.

[4 marks]

The driver program, **FerryLoadChecker**, accepts *number plate* and *weight (kg)* inputs from the user and computes the *total weight (kg)*. The panel should display all the labels shown, one textbox on top to display the total weight, a textarea to show all entries line by line, one textbox to input number plate, one textbox to input weight, two buttons to add different type of vehicle for the classes created above, and one "RESET" button to clear all entries and zeroize the total weight (*see Figure Q2.1*).

[12 marks]

Handle the error exception and display the error message "Invalid input" dialog box if the user inputs non-double into the weight textbox and hits any of the add buttons.

[1 mark]

Continued...

Ferry Load Checker

Total Weight (kg) 10786.0

Car WQA9017 = 2021.0kg

Truck RT1112 = 8765.0kg

Number Plate RT1112 Weight (kg) 8765

Add car RESET

*Figure Q2.1***Continued...**

Question 3

Create a 4-digit keypad GUI program for a safe.

The program should have all the components in the following layout (see *Figure Q3.1*):-

- Textbox to display user input
- 0 ~ 9 digit buttons
- "LOCK" button
- "CLEAR" button

[Note: with ActionListener interface and actionPerformed method]

[9 marks]

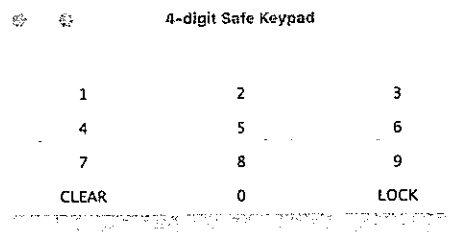


Figure Q3.1

The user needs to click on the digit button to input the pin code. The pin code will be displayed in the textbox for every click (see *Figure Q3.2*). An error dialog box will be displayed with the message "Only 4 digits!" (see *Figure Q3.3*) if the user clicks more than 4 digits.

[2 marks]

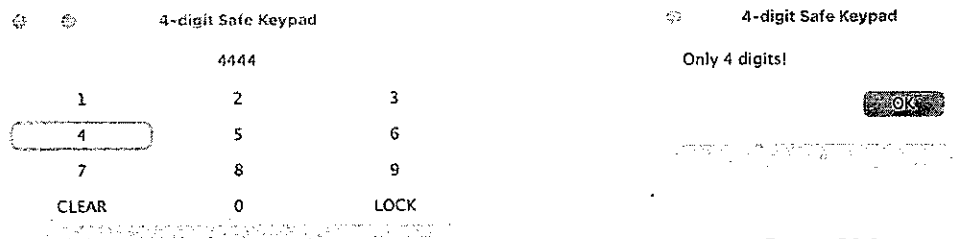


Figure Q3.2

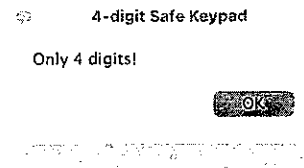


Figure Q3.3

Once the user has confirmed the pin code, the user is required to click on the "LOCK" button to lock it. A dialog box will be displayed with the message "LOCKED" (see *Figure Q3.4*). An error dialog box will be displayed with the message "Can only be locked with 4 digits!" if the user tries to lock with less than 4 digits (see *Figure Q3.5*). Display "It is locked!" message in a dialog box if the user click on the "LOCK" button again after a successful lock (see *Figure Q3.6*).

[7 marks]

Continued...

4-digit Safe Keypad



LOCKED

OK

Figure Q3.4

4-digit Safe Keypad



Can only be locked with 4 digits!

OK

Figure Q3.5

4-digit Safe Keypad



It is locked!

OK

Figure Q3.6

The user is able to click on the “CLEAR” button to clear their input and empty the textbox display.

[1 mark]

To unlock, the user just needs to click on the digit button with the correct pin code. A dialog box with the message “UNLOCKED” (see *Figure Q3.7*) will be displayed if the pin code is correct. Otherwise, an error dialog box with the message “Invalid PIN!” (see *Figure Q3.8*) will be displayed.

[6 marks]

4-digit Safe Keypad



UNLOCKED

OK

Figure Q3.7

4-digit Safe Keypad



Invalid PIN!

OK

Figure Q3.8

